Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

CE218139 – PSoC 6 MCU with BLE Connectivity: Eddystone Beacon (RTOS)

Objective

This code example demonstrates a Bluetooth Low Energy (BLE) beacon that broadcasts the core frame types (UID, URL, and TLM) of Google's Eddystone beacon profile.

Overview

This code example demonstrates the ability of PSoC® 6 MCU with BLE Connectivity (PSoC 6 MCU) to function as a BLE beacon using the Broadcaster role, which transmits Eddystone fames. Eddystone is an open-source BLE beacon profile released by Google. This project broadcasts core Eddystone frame types—Eddystone UID, Eddystone URL, and Eddystone TLM.

This code example assumes that you are familiar with the PSoC 6 MCU and the PSoC Creator™ Integrated Design Environment (IDE). If you are new to PSoC 6 MCU, see the application note AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity.

This code example uses FreeRTOS. See PSoC 6 101: Lesson 1-4 FreeRTOS training video to learn how to create a PSoC 6 FreeRTOS project with PSoC Creator. Visit the FreeRTOS website for documentation and API references of FreeRTOS.

Note: This project requires an Android device with Android 5.0 or a later version to evaluate.

Requirements

Tool: PSoC Creator 4.2; Peripheral Driver Library (PDL) 3.0.1

Programming Language: C (Arm® GCC 5.4.1)

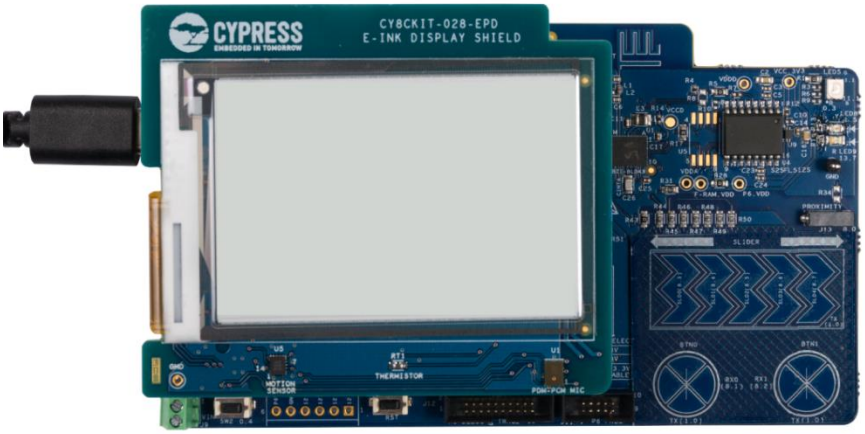
Associated Parts: All PSoC 6 MCUs with BLE Connectivity

Related Hardware: CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit

Hardware Setup

Plug in the E-INK display shield on to the Pioneer Board as Figure 1 shows.

Figure 1. Hardware Setup



Set the switches and jumpers on the Pioneer Board as shown in Table 1.

Table 1. Switch and Jumper Selection

Switch/Jumper	Position	Location
SW5	3.3 V	Front
SW6	PSoC 6 BLE	Back
SW7	V_{DDD} / KitProg2	Back
J8	Installed	Back

Software Setup

Install the CY8CKIT-62-BLE PSoC 6 BLE Pioneer Kit software, which contains all the required software to evaluate this code example. No additional software setup is required.

Operation

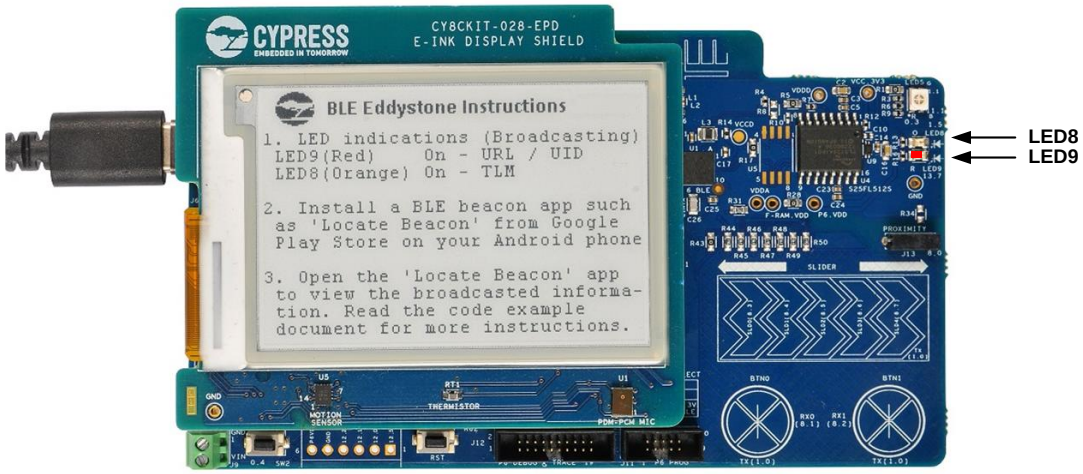
Note: This code example requires an Android device with Android 5.0 or a later version to evaluate.

1. Install a BLE beacon application from Google Play Store that supports Eddystone profile. Locate Beacon, which is the recommended application for this code example, is used to demonstrate the project operation in this section.
2. Power the Pioneer Baseboard through the USB connector **J10**.
3. Program the Pioneer Baseboard with the *CE220186_BLE_Eddystone* project. See the Pioneer Kit guide for details on how to program firmware into the device.

   After programming successfully, the E-INK display will refresh and show the instructions to use this project. BLE will start broadcasting URL frames with interleaved TLM frames. You can change the Eddystone settings by editing the *eddystone_config.h* header file.

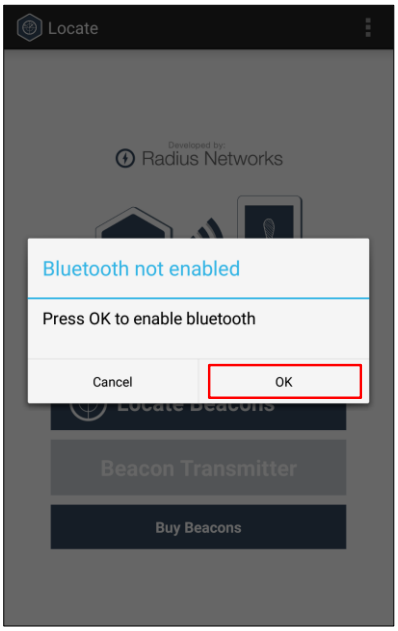
   The red LED (**LED9**) remains ON during the broadcast of URL/UID frames and the orange LED (**LED8**) remains ON during the broadcast of TLM frames.

Figure 2. BLE Broadcasting



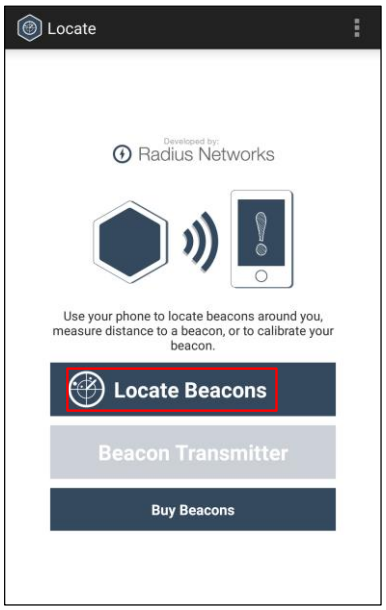
4. Open the Locate Beacon app on the mobile device. If Bluetooth is not enabled on the device, the application will prompt to enable it.

Figure 3. Enabling Bluetooth



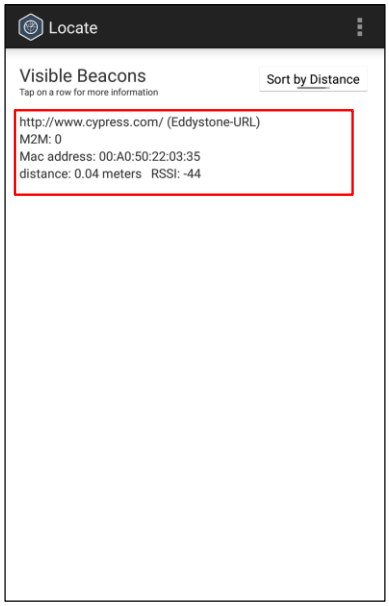
5. After Bluetooth is enabled, select the **Locate Beacons** option as shown in Figure 4.

Figure 4. Locating Eddystone Beacon



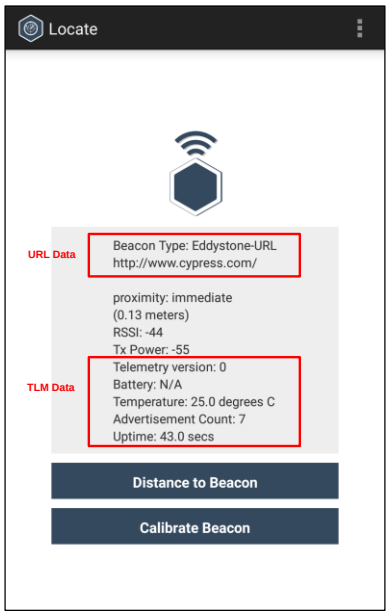
The application will search for available beacons and list them. Select the beacon that broadcasts http://www.cypress.com/ in Eddystone-URL mode, as Figure 5 shows.

Figure 5. Selecting the BLE Eddystone Beacon



6. After selecting the beacon, the application will continuously refresh the screen with the URL and TLM frames broadcasted by the PSoC 6 MCU, as Figure 6 shows.

Figure 6. Viewing Eddystone Data



Design and Implementation

This code example supports the following Eddystone core frame types:

- **Eddystone-UID** broadcasts a unique, static ID with a 10-byte Namespace field and a 6-byte Instance field.
- **Eddystone-URL** broadcasts a compressed URL that, once parsed and decompressed, is directly usable by the client.
- **Eddystone-TLM (unencrypted)** broadcasts information about the beacon. This can include beacon uptime, number of packets transmitted, battery level, beacon temperature etc. The TLM frame should be interleaved with an identifying frame such as Eddystone-UID or Eddystone-URL.

For more information on Eddystone profile and frame formats, see the official Eddystone GitHub page.

Figure 7, Figure 8 and Figure 9 show the TopDesign schematic of this code example. The BLE Component is configured for non-discoverable broadcaster role that transmits Eddystone frames as non-connectable undirected advertisement packets. You can select one of UID and URL frames as a compile-time option, using the macros in the *eddystone_config.h* header file. The code example broadcasts URL frames by default with TLM frames periodically interleaved between URL frames. The frame timings can be adjusted as another compile-time option available in the *eddystone_config.h* header file.

Two LEDs on the Pioneer Board are used to indicate the current frame being broadcast. The red LED (**LED9**) remains ON during the broadcast of URL/UID frames and the orange LED (**LED8**) remains ON during the broadcast of TLM frames.

The MCWDT Component is configured to create interrupts at 100 ms time intervals. These time intervals are used to track the uptime (time elapsed since power-on or reset). The uptime data is used in TLM frame.

The E-INK display shows the instructions to use this code example at startup and is then turned OFF to save power. E-INK displays consume no power to retain the display. For more details on E-INK display, see the code example CE218136 – PSoC 6 MCU E-INK Display with CapSense (RTOS).

Figure 7. TopDesign Schematic: BLE, MCWDT, and LEDs

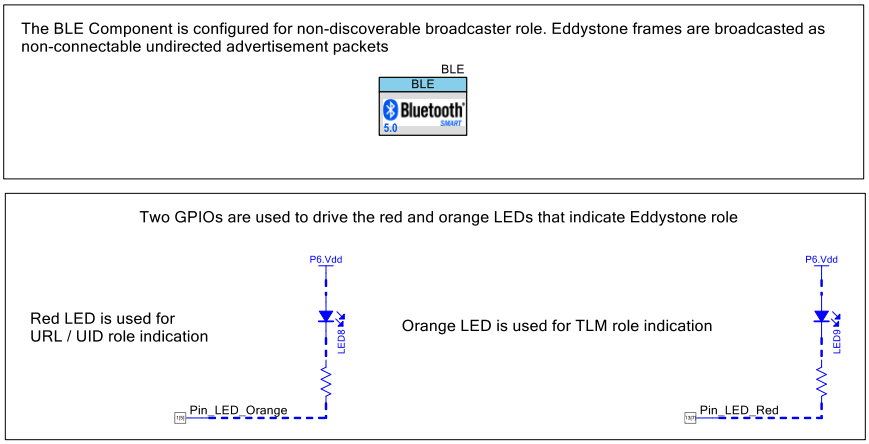


Figure 8 TopDesign Schematic: E-INK Display

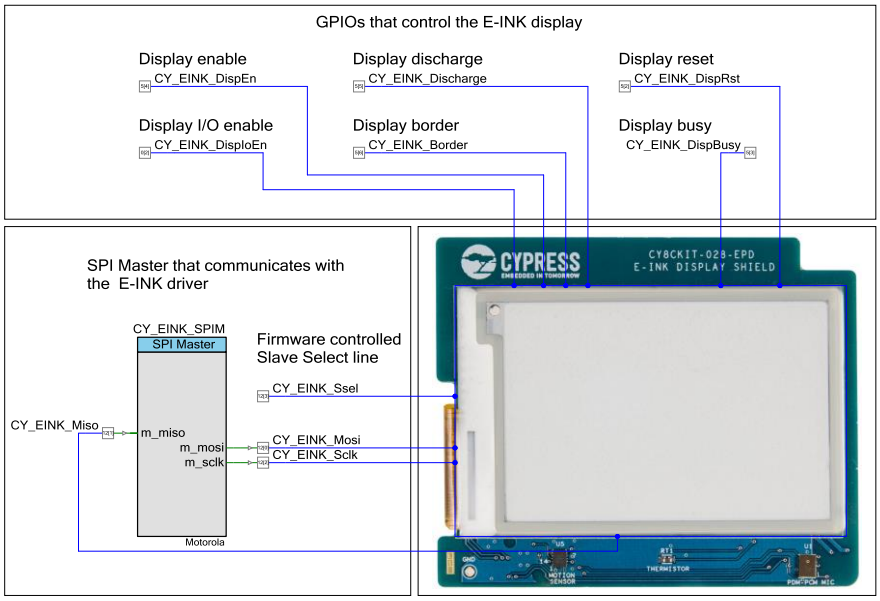
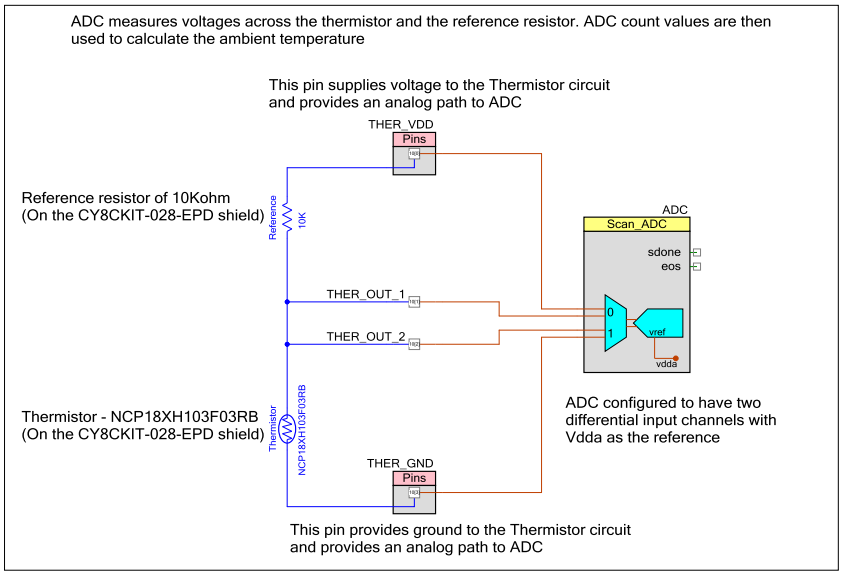


Figure 9. TopDesign Schematic: Temperature Compensation for E-INK Display



The code example consists of the following files:

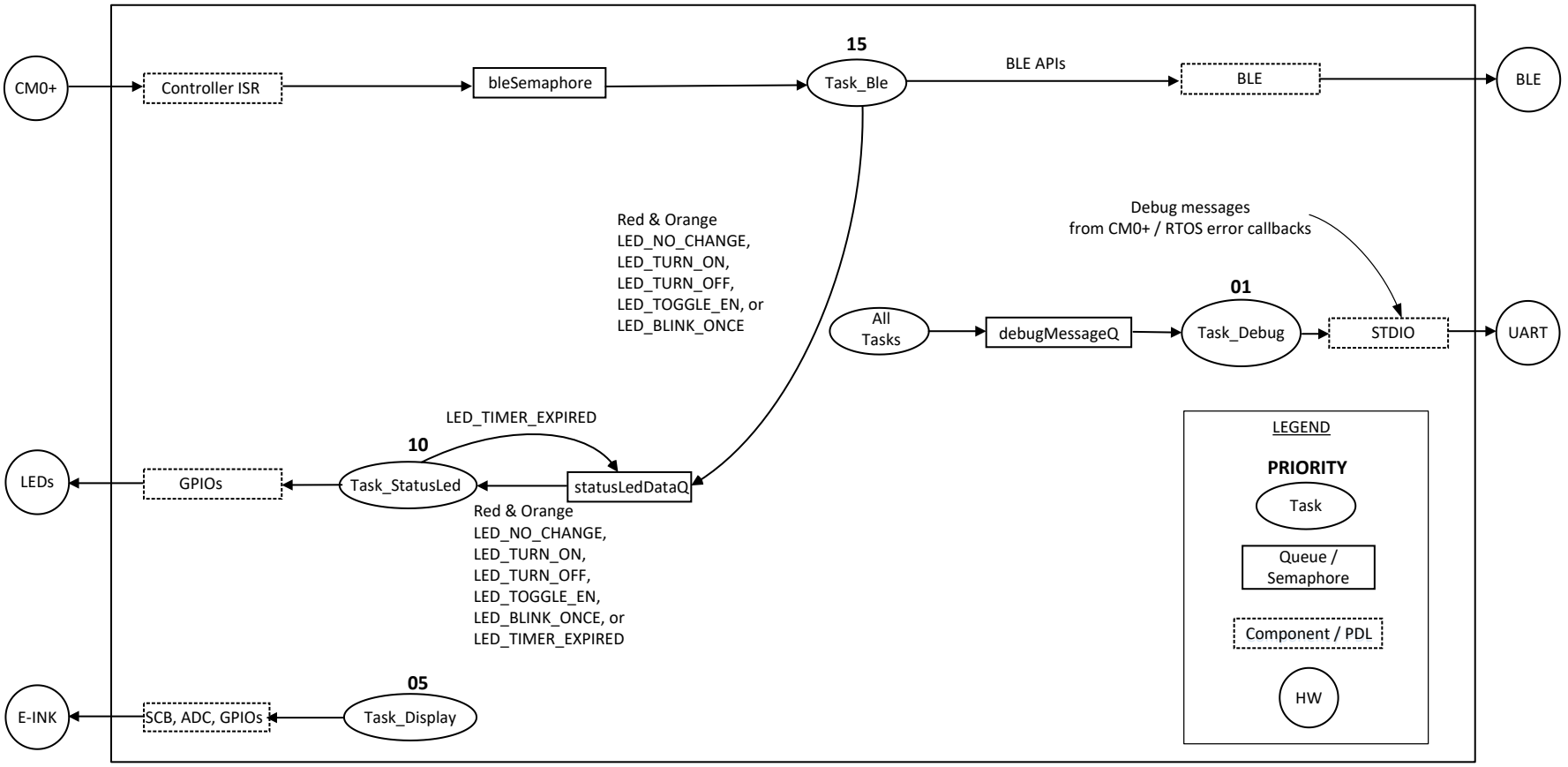
- *FreeRTOSConfig.h* contains the FreeRTOS settings and configuration. Non-default settings are explained with in-line comments.
- *main_cm4.c* contains the main function, which is the entry point and execution of the firmware application. The main function sets up user tasks and then starts the RTOS scheduler.
- *main_cm0p.c* contains functions that start up the BLE controller, start up the CM4, and continuously service BLE stack events.
- *ble_task.c/.h* contain the task and associated functions that handle BLE beacon broadcast.
- *eddystone_config.h* contains the macros that configure Eddystone frame details.
- *status_led_task.c/h* – contain the task that controls status LED indications.
- *display_task.c/.h* contain the task that initialize the E-INK display and show the instructions to use code example at startup[1].
- *uart_debug.c/h* contain the task and functions that enable UART based debug message printing.
- *screen_contents.c/h* contain the text and background images used by the display module.
- *temperature_eink.c/h* contain functions that measure ambient temperature for E-INK display compensation

See the corresponding header/source files for more details.

Figure 10 shows the RTOS firmware flow of this code example.

[1] For a detailed list of files included in the E-INK Library, see the code example, CE218136 – PSoC 6 MCU E-INK Display with CapSense (RTOS)

Figure 10. RTOS Firmware Flow



Components

Table 2. List of PSoC Creator Components

Component	Instance Name	Function
BLE	BLE	The BLE Component is configured as a non-discoverable broadcaster role that transmits Eddystone frames as non-connectable undirected advertisement packets.
Digital Output Pin	Pin_LED_Red Pin_LED_Orange	These GPIOs are configured as firmware controlled digital output pins that control status LEDs.
UART	DEBUG_UART	UART is used to transmit debug information to a terminal (disabled by default)

Note: See the code example CE218136 – PSoC 6 MCU E-INK Display with CapSense (RTOS) for more details on components used by E-INK library and temperature compensation.

See the PSoC Creator project for more details of PSoC Component configurations and design wide resource settings.

Related Documents

Application Notes	
AN210781 – Getting Started with PSoC 6 MCU with Bluetooth Low Energy (BLE) Connectivity	Describes PSoC 6 MCU with BLE Connectivity devices and how to build your first PSoC Creator project
AN215656 – PSoC 6 MCU: Dual-Core CPU system Design	Describes the dual-core CPU architecture in PSoC 6 MCU, and shows how to build a simple dual-core design
AN219434 – Importing PSoC Creator Code into an IDE for a PSoC 6 MCU Project	Describes how to import the code generated by PSoC Creator into your preferred IDE
PSoC Creator Component Datasheets	
Pins	Supports connection of hardware resources to physical pins
Bluetooth Low Energy	Supports BLE connectivity.
Device Documentation	
PSoC 6 MCU: PSoC 63 with BLE Datasheet	PSoC 6 MCU: PSoC 63 with BLE Architecture Technical Reference Manual
Development Kit Documentation	
CY8CKIT-062-BLE PSoC 6 BLE Pioneer Kit	
Training Videos	
PSoC 6 101: Lesson 1-4 FreeRTOS	

Document History

Document Title: CE218139 – PSoC 6 MCU with BLE Connectivity: Eddystone Beacon (RTOS)

Document Number: 002-18139

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	6096842	NIDH	03/13/2018	Initial public release version

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Arm® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6 MCU

Cypress Developer Community

Community Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2018. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.